جامعة المستقبل

قسم الأنظمة الطبية الذكية

Application Development II

3rd Class, Second Semester.

(2024-2025)

WHAT IS FLUTTER?

Flutter allows you to build beautiful native apps on iOS and Android from a single codebase

- Open-source mobile app SDK
- Developed by Google
- Building high-performance apps for iOS and Android, from a single codebase

WHY USE FLUTTER?

Flutter makes it easy and fast to build beautiful mobile apps.

- Reactive framework
- Material and Cupertino widgets
- Hot reload
- Dart language and core libs
- Interop with mobile SDKs
- Android Studio/IntelliJ official IDE
- Debugger, Format


Framework
(Dart)
Material
Cupertino
Widgets
Rendering
Animation
Painting
Gestures
Foundation
Engine (C++)
Skia
Dart
Text

DART LANGUAGE

Productive. Syntax must be clear and concise, tooling simple.

• Fast. Runtime performance and startup must be great and predictable even on small mobile devices.

• Portable. Client developers have to think about three platforms today: iOS, Android, and Web. The language needs to work well on all of them.

• Approachable. The language can't stray too far from the familiar if it wishes to be relevant for millions of developers.

• Reactive. A reactive style of programming should be supported by the language.

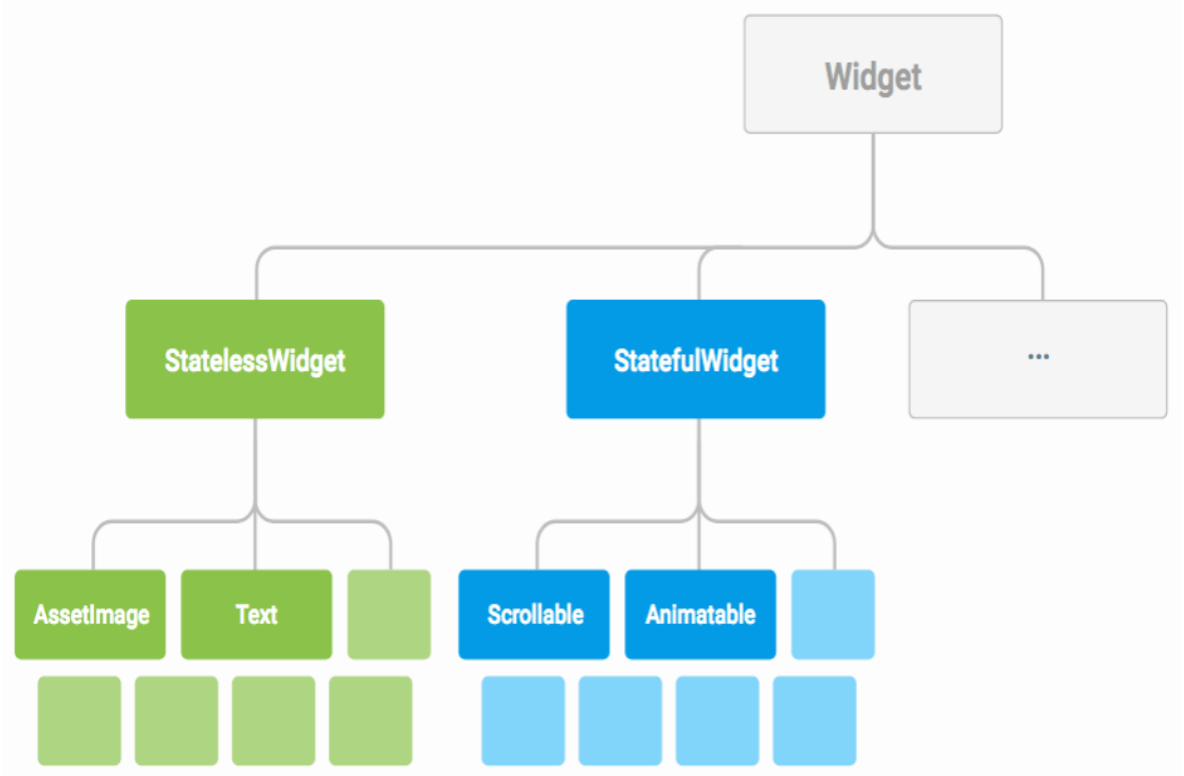
Widgets are the basic building blocks. Each widget is an immutable declaration of part of the user interface. Unlike other frameworks that separate views, view controllers, layouts, and other properties, Flutter has a consistent, unified object model: the widget.

A widget can define:

● a structural element (like a button or menu)

● a stylistic element (like a font or color scheme)

● an aspect of layout (like padding)



Example

```
main() {
runApp( new MaterialApp
        ( title: 'Flutter Demo', theme: new ThemeData
          ( primarySwatch: Colors.green, ),
home: new Scaffold( appBar: new AppBar
                    ( title: new Text("Flutter Demo"),
),
body: new Text("Hello World!"),
), ),)
```

Stateless Widget

```
new Text( 'Hello! How are you?', textAlign: TextAlign.center, overflow: TextOverflow.ellipsis, style:
new TextStyle(fontWeight: FontWeight.bold), )
```

In short...

Variables & constants:

```
var name = 'Voyager I';
var year = 1977;
final bar = const [];
const baz = const [];
```

Getters and setters

```
class Spacecraft {
   DateTime launchDate ;
   int get launchYear => launchDate?.year ;
}
```

Functions:

```
int fibonacci(int n) {
 if (n == 0 || n == 1) return n;
 return fibonacci(n - 1) + fibonacci(n - 2);
}
```

Lambda (fat-arrow) expressions:

```
flybyObjects.where((name) => name.contains('flower')).forEach(print);
```